Compartilhamento de Recursos entre Origens Diferentes (CORS)

CORS (Cross-Origin Resource Sharing — Compartilhamento de Recursos entre Origens Diferentes) é um **mecanismo baseado em cabeçalho HTTP** que permite que um servidor indique quaisquer **origens (domínios, esquemas ou portas)** diferentes da sua própria a partir das quais um navegador deve permitir o carregamento de recursos.

Um navegador web faz uma requisição **cross-origin** (entre origens diferentes) quando requisita um recurso de um domínio, protocolo ou porta diferente daquele do documento atual.
Um exemplo típico é um script hospedado em https://dominio-a.com que faz uma requisição a https://dominio-b.com/dados.json.

CORS é usado para relaxar a política de mesma origem (SOP), que restringe como um documento ou script carregado de uma origem pode interagir com recursos de outra origem.

Por que o CORS é necessário?

A política de mesma origem é a política de segurança fundamental da web, que impede que documentos e scripts carregados de uma origem acessem recursos de outra origem.
Sem essa política, qualquer site malicioso poderia fazer requisições para APIs sensíveis do lado do usuário (como dados bancários, redes sociais etc.).

Entretanto, essa política é bastante restritiva e torna difícil compartilhar recursos entre diferentes domínios legítimos (como entre api.exemplo.com e app.exemplo.com).

O CORS fornece um meio **seguro e flexível** para os servidores controlarem quem pode acessar seus recursos.

Requisições simples

Algumas requisições são classificadas como **requisições simples**.
Elas devem atender a todas as seguintes condições:

1. Usam um dos métodos: GET, HEAD ou POST.
2. Os cabeçalhos personalizados são limitados a:
    - Accept
    - Accept-Language
    - Content-Language
    - Content-Type (com valor limitado a application/x-www-form-urlencoded, multipart/form-data ou text/plain)
3. Não usam objetos ReadableStream como corpo.

Nesses casos, o navegador **não faz uma requisição prévia (preflight)**, e apenas envia a requisição diretamente com um cabeçalho adicional Origin.

O servidor pode então responder com um cabeçalho:

Ou, para permitir acesso universal:

```http
Access-Control-Allow-Origin: *
```

⚠️ Quando Access-Control-Allow-Origin é *, **as credenciais (cookies, autenticação HTTP) não são incluídas**.

Requisições com preflight

Requisições que **não atendem aos critérios de requisição simples** exigem um processo adicional chamado de **preflight** (pré-verificação).

O navegador primeiro envia uma requisição OPTIONS com os seguintes cabeçalhos:

- Origin: origem da requisição
- Access-Control-Request-Method: método HTTP pretendido (ex: PUT)
- Access-Control-Request-Headers: cabeçalhos personalizados pretendidos

Se o servidor aceitar a requisição, responde com:

```http
Access-Control-Allow-Origin: https://dominio-a.com
Access-Control-Allow-Methods: PUT, DELETE
Access-Control-Allow-Headers: X-Custom-Header
```

Se a resposta for aceitável, **o navegador envia a requisição original**.

Requisições com credenciais

Por padrão, **requisições cross-origin não enviam cookies ou cabeçalhos de autenticação HTTP**.

Para incluir credenciais:

- O cliente deve definir credentials: "include" na requisição.
- O servidor **deve responder** com:

```http
Access-Control-Allow-Credentials: true
```

- Além disso, **Access-Control-Allow-Origin não pode ser *** — deve indicar uma origem específica.

Cache de preflight

A resposta da requisição preflight pode ser armazenada em cache por um tempo, definido pelo servidor com:

```http
Access-Control-Max-Age: 600
```

Isso evita que o navegador envie uma nova requisição OPTIONS para cada requisição real subsequente.

Exemplo prático

Requisição de um script em https://exemplo.com para https://api.outraorigem.com com método PUT:

javascript Copiar Editar

```javascript
fetch("https://api.outraorigem.com/dados", {
  method: "PUT",
  headers: {
    "Content-Type": "application/json",
    "X-Custom-Header": "valor"
  },
  body: JSON.stringify({ usuario: "João" }),
  credentials: "include"
});
```

Requisição preflight:

http Copiar Editar

```http
OPTIONS /dados HTTP/1.1
Origin: https://exemplo.com
Access-Control-Request-Method: PUT
Access-Control-Request-Headers: X-Custom-Header, Content-Type
```

Resposta do servidor:

http Copiar Editar

```http
Access-Control-Allow-Origin: https://exemplo.com
Access-Control-Allow-Methods: PUT
Access-Control-Allow-Headers: X-Custom-Header, Content-Type
Access-Control-Allow-Credentials: true
```

Cabeçalhos CORS

Cabeçalho	Direção	Descrição
`Origin`	Requisição	Origem da requisição
`Access-Control-Allow-Origin`	Resposta	Origens permitidas
`Access-Control-Allow-Credentials`	Resposta	Permite envio de cookies/autenticação
`Access-Control-Allow-Headers`	Resposta	Cabeçalhos permitidos na requisição
`Access-Control-Allow-Methods`	Resposta	Métodos permitidos
`Access-Control-Expose-Headers`	Resposta	Cabeçalhos visíveis ao JavaScript do cliente
`Access-Control-Max-Age`	Resposta	Tempo (em segundos) para cache da resposta preflight
`Access-Control-Request-Headers`	Requisição	Lista de cabeçalhos pretendidos
`Access-Control-Request-Method`	Requisição	Método pretendido